

COMSC-210 Lab 15

Graph Searches

GOOD PROGRAMMING PRACTICES [show / hide](#)

ABOUT THIS ASSIGNMENT

In this lab assignment you will develop a graph-based solution for a roadmap, the data for which is stored in a text file of city names and distances between adjacent cities. You will use BFS and DFS to search out all the nodes in a graph, starting from a user-specified city

Abstract data types will *not* be used. For the route calculation you will use the [STL list](#) template for the adjacency lists. We'll also make use of the [STL queue](#), the [STL stack](#), and the [STL pair](#).

After you complete this lab assignment, post the required files to the [COMSC server](#) so that you receive the proper credit for this 50-point lab. Your posted files are due at midnight of the evening of the due date indicated in the course outline. Use the "Submit your FA2015 work" link on the class website to post your file for this lab, using the **lab16** folder provided.

LAB 15a: Breadth First Search, BFS [GraphSearchBFS.cpp]

Write **GraphSearchBFS.cpp**, by completing the supplied CPP file. Right-click [here](#) to download the supplied file that contains the main function and a function outline for BFS. Right-click [here](#) to download the input file to use, containing road map information for California. Click [here](#) for a map of California. Write the missing code block in the missing BFS function, based on the algorithm in the lecture notes.

NOTE: I want you to eliminate "whitespace" from the values read from the input TXT file. Users will skip whitespace in their input -- e.g., "San Francisco" will be entered as "SanFrancisco". So after reading `fromCity` and `toCity`, do this:

```
fromCity.erase(remove_if(fromCity.begin(), fromCity.end(), ::isspace), fromCity.end());
toCity.erase(remove_if(toCity.begin(), toCity.end(), ::isspace), toCity.end());
```

Submit the CPP to the class website for credit. Do NOT submit the TXT file.

Program I/O. Input: As prompted, a start city. Output: The BFS-based list of cities reachable from start.

Program I/O. Here's what the result should be for the start city of San Francisco:

Enter the start city [blank to exit]: SanFrancisco

BFS-SanFrancisco-Eureka-Redding-Sacramento-Fresno-Bakersfield-SanLuisObispo-CoosBay-GrantsPass-KlamathFalls-Lakeview-Reno-Bishop-Barstow-LosAng

LAB 15b: Depth First Search, DFS [GraphSearchDFS.cpp]

Write **GraphSearchDFS.cpp**, by modifying the lab 15a CPP. Replace the `doBreadthFirstSearch` BFS function with `doDepthFirstSearch`, doing DFS based on the algorithm in the lecture notes.

Remember to use *reverse iteration* where it's indicated in the DFS algorithm, and remember that `iterator::end` is *not* a valid node! There are several ways to do this, and `for (it=x.end(); it!=x.begin(i); it--)` is *not* one of them.

Submit the CPP to the class website for credit. Do NOT submit the TXT file.

Program I/O. Input: As prompted, a start city. Output: The DFS-based list of cities reachable from start.

Program I/O. Here's what the result should be for the start city of San Francisco:

Enter the start city [blank to exit]: SanFrancisco

DFS-SanFrancisco-Eureka-CoosBay-GrantsPass-Redding-KlamathFalls-Lakeview-Reno-Sacramento-Bishop-Winnemucca-Ely-LasVegas-SanDiego-Barstow-Bakers

How to pause a console program: [show / hide](#)

GRADING RUBRIC [show/hide](#)
